Backpropagation Through Agents

Zhiyuan Li[1] Wenshuai Zhao[2] Lijun Wu[1] Joni Pajarinen[2]

[1] School of Computer Science and Engineering, University of Electronic Science and Technology of China

[2] Department of Electrical Engineering and Automation, Aalto University

Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

Abstract

A fundamental challenge in multi-agent reinforcement learning (MARL) is to learn the joint policy in an extremely large search space, which grows exponentially with the number of agents. Moreover, fully decentralized policy factorization significantly restricts the search space, which may lead to sub-optimal policies. In contrast, the auto-regressive joint policy can represent a much richer class of joint policies by factorizing the joint policy into the product of a series of conditional individual policies. While such factorization introduces the action dependency among agents explicitly in sequential execution, it does not take full advantage of the dependency during learning. In particular, the subsequent agents do not give the preceding agents feedback about their decisions. In this paper, we propose a new framework Back-Propagation Through Agents (BPTA) that directly accounts for both agents' own policy updates and the learning of their dependent counterparts. This is achieved by propagating the feedback through action chains. With the proposed framework, our Bidirectional Proximal Policy Optimisation (BPPO) outperforms the state-of-the-art methods. Extensive experiments on matrix games, StarCraftII v2, Multi-agent MuJoCo, and Google Research Football demonstrate the effectiveness of the proposed method.

1 Conditional MAPG

Theorem 1 (Conditional Multi-Agent Stochastic Policy Gradient Theorem)

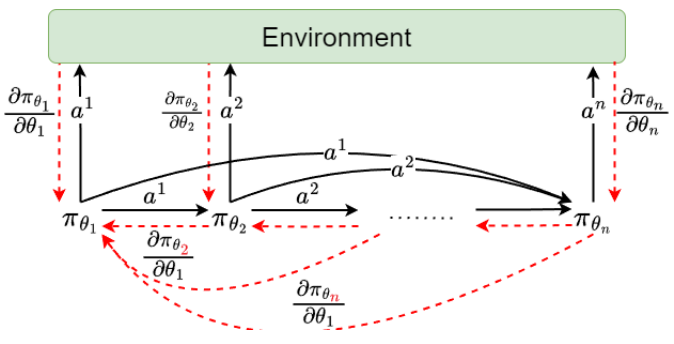
For any episodic cooperative stochastic game with n agents, the gradient of the expected total reward for agent i, who has a backward dependency on some other peer agents B_i using parameters θ_{B_i}, with respect to current policy parameters θ_i is:

$$\nabla_{\theta_i} J(\boldsymbol{\theta}) = \int_S \rho^{\boldsymbol{\pi}}(s) \Big[\int_{A^i} \underbrace{\nabla_{\theta_i} \pi_{\theta_i}(a^i|s, a^{F_i})}_{\text{Own Learning}} \int_{A^{-i}} \pi_{\theta_{-i}}(a^{-i}|s', a^{F_{-i}}) + \int_{A^i} \pi_{\theta_i}(a^i|s, a^{F_i}) \int_{A^{F_i}} \pi_{\theta_{F_i}}(a^{F_i}|s, a^{F_{F_i}}) \int_{A^{B_i}} \underbrace{\nabla_{a^i} \pi_{\theta_{B_i}}(a^{B_i}|s, a^i, a^{F_{B_i}\setminus\{i\}}) \nabla_{\theta_i} g(\theta_i, \varepsilon)}_{\text{Peer Learning}} \Big] Q_{\boldsymbol{\pi}}(s, \boldsymbol{a}) d\boldsymbol{a}^{-i} da^i ds, \tag{1}$$

where F_{B_i} indicates the set of agents on which B_i have forward dependencies.

We note that the policy gradient for agent *i* at each state has two primary terms. The first term $\nabla_{\theta_i} \pi_{\theta_i}(a^i|s, a^{F_i})$ corresponds to the independent multi-agent policy gradient which explicitly differentiates through π_{θ_i} with respect to the current parameters θ_i. This enables agent *i* to model its own learning. By contrast, the second term $\nabla_{a^i} \pi_{\theta_{B_i}}(a^{B_i}|s, a^i, a^{F_{B_i}\setminus\{i\}}) \nabla_{\theta_i} g(\theta_i, \varepsilon)$ aims to additionally account for how the consequences of the corresponding action on its backward dependent agents' policies influence its direction of performance improvement.

2 Bidirectional Proximal Policy Optimisation



Our proposed algorithm can be conveniently integrated into most PG-based methods. Given the empirical performance and monotonic policy improvement of PPO, we propose *Bidirectional Proximal Policy Optimisation* (BPPO) to incorporate the proposed theorem with PPO.

University of Electronic Science and Technology of China

3 Experiments

To validate BPPO, we conduct extensive experiments on several multi-agent benchmarks, including two matrix games, the StarCraft Multi-Agent Challenge Version 2 (SMACv2), the Multi-agent MuJoCo (MA-MuJoCo), and the Google Research Football (GRF). Results show that BPPO improves the performance against current state-of-the-art MARL methods.

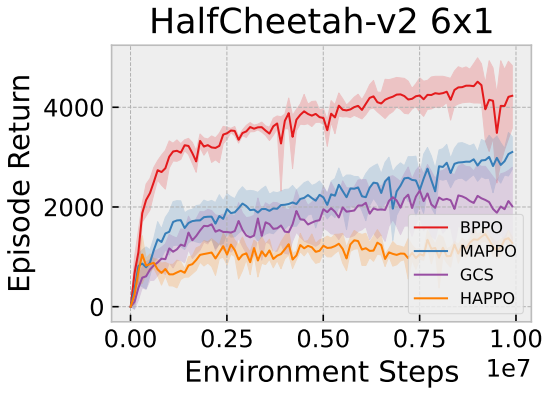


Figure 1: Performance comparison on multiple Multi-Agent MuJoCo tasks.

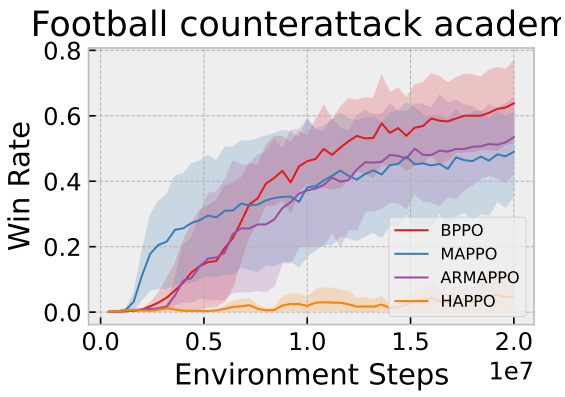


Figure 2: Averaged train win rate on the Google Research Football scenarios.

GitHub

A digital version of this presentation can be found here:
`https://github.com/LiZhYun/BackPropagationThroughAgents`.
In case your audience finds it hard to remember this url, here is a QR code generated by LaTeX: